**Ques 1 : Create authentication service that returns JWT**

**-- AuthController.java**

package com.example.jwtauthservice.controller;  
  
import com.example.jwtauthservice.util.JwtUtil;  
import jakarta.servlet.http.HttpServletRequest;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.\*;  
import org.springframework.web.bind.annotation.\*;  
  
import java.nio.charset.StandardCharsets;  
import java.util.\*;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(HttpServletRequest request) {  
 String authHeader = request.getHeader("Authorization");  
  
 if (authHeader == null || !authHeader.startsWith("Basic ")) {  
 return ResponseEntity.status(HttpStatus.UNAUTHORIZED)  
 .header("WWW-Authenticate", "Basic realm=\"JWT Auth\"")  
 .body(Collections.singletonMap("error", "Authentication failed: Missing or invalid Authorization header"));  
 }  
  
 try {  
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);  
 String credentials = new String(credDecoded, StandardCharsets.UTF\_8);  
  
 String[] values = credentials.split(":", 2);  
 if (values.length != 2) {  
 return ResponseEntity.status(HttpStatus.BAD\_REQUEST)  
 .body(Collections.singletonMap("error", "Invalid credentials format"));  
 }  
  
 String username = values[0];  
 String password = values[1];  
  
 if (!"user".equals(username) || !"pwd".equals(password)) {  
 return ResponseEntity.status(HttpStatus.UNAUTHORIZED)  
 .body(Collections.singletonMap("error", "Invalid username or password"));  
 }  
  
 String token = jwtUtil.generateToken(username);  
 return ResponseEntity.ok(Collections.singletonMap("token", token));  
  
 } catch (IllegalArgumentException e) {  
 return ResponseEntity.status(HttpStatus.BAD\_REQUEST)  
 .body(Collections.singletonMap("error", "Failed to decode credentials"));  
 }  
 }  
}

**-- JwtUtil.java**

package com.example.jwtauthservice.util;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import org.springframework.stereotype.Component;  
  
import java.util.Date;  
  
@Component  
public class JwtUtil {  
  
 private final String SECRET\_KEY = "mysecretkey";  
  
 public String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.currentTimeMillis()))  
 .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 10)) // 10 minutes  
 .signWith(SignatureAlgorithm.HS256, SECRET\_KEY)  
 .compact();  
 }  
}

**--SecurityConfig.java**

package com.example.jwtauthservice.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.Customizer;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").permitAll()  
 .anyRequest().authenticated()  
 )  
 .httpBasic(Customizer.withDefaults());  
  
 return http.build();  
 }  
}